

## BEST AVAILABLE COPY

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-067403  
 (43)Date of publication of application : 07.03.2003

(51)Int.CI. G06F 17/30  
 G06F 17/21

(21)Application number : 2001-255016

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 24.08.2001

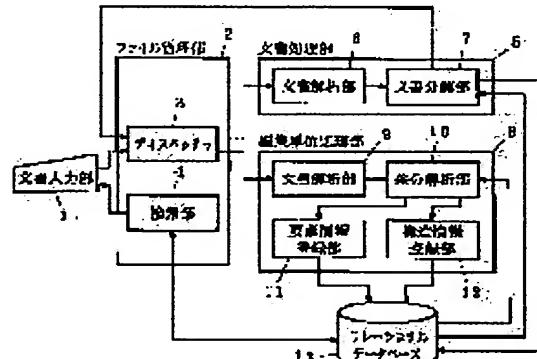
(72)Inventor : NUMATA KENICHI  
 KAWABE SHIGEHISA  
 NUKAGA MASAO  
 YAMADA SUEFUMI  
 IKEDA MINORU  
 AZUMA KAZUHIKO  
 YAMADA YOSHIO

**(54) STRUCTURED DOCUMENT MANAGING DEVICE, METHOD THEREFOR, RETRIEVING DEVICE AND METHOD THEREFOR**

**(57)Abstract:**

**PROBLEM TO BE SOLVED:** To provide a structured document managing device which can provide an environment where more than one person can jointly edit the document, and a method therefor, and to provide a retrieving device which efficiently determine the ancestor-descendant relation of a structured document and quickly retrieves the structure of the document, and a method therefor.

**SOLUTION:** A structured document inputted from a document inputting part 1 is decomposed into partial structures which become editing units, at a document processing part 5, and global-structure information showing the relation between the partial structures is registered with a relational database 13. The partial structures thus obtained are inputted to an edit unit processing part 8, and information on each element as element information and the relation between respective elements in the partial structures as the structural information are registered with the relational database 13. The ancestor-descendant relation between two elements can simply and quickly be found by the decision on the ancestor-descendant relation between the partial structures by the global structural information and the ancestor-descendant relation by the structural information on the partial structures on the part of the ancestors.



**LEGAL STATUS**

[Date of request for examination] 24.09.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-67403

(P2003-67403A)

(43)公開日 平成15年3月7日(2003.3.7)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テ-マコード(参考)
G 0 6 F 17/30	2 3 0	G 0 6 F 17/30	2 3 0 Z 5 B 0 0 9
	1 4 0		1 4 0 5 B 0 7 5
	1 8 0		1 8 0 D
17/21	5 0 1	17/21	5 0 1 T

審査請求 未請求 請求項の数8 OL (全18頁)

(21)出願番号	特願2001-255016(P2001-255016)	(71)出願人 000005496 富士ゼロックス株式会社 東京都港区赤坂二丁目17番22号
(22)出願日	平成13年8月24日(2001.8.24)	(72)発明者 沼田 賢一 神奈川県横浜市西区みなとみらい3丁目3番1 富士ゼロックス株式会社内
		(72)発明者 川邊 康久 神奈川県横浜市西区みなとみらい3丁目3番1 富士ゼロックス株式会社内
		(74)代理人 100101948 弁理士 柳澤 正夫

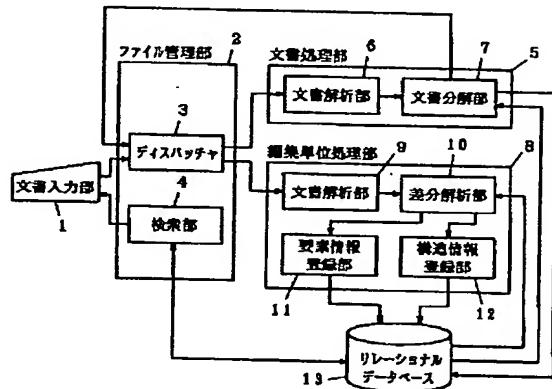
最終頁に統ぐ

(54)【発明の名称】 構造化文書管理装置及び構造化文書管理方法、検索装置、検索方法

(57)【要約】

【課題】 複数人が共同して編集可能な環境を提供できる構造化文書管理装置及び方法と、構造化文書中の要素の先祖・子孫関係を効率よく判定して文書構造の検索を高速に行う検索装置及び検索方法を提供する。

【解決手段】 文書入力部1から入力された構造化文書は、文書処理部5で編集単位となる部分構造に分解され、部分構造間の関係を示すグローバル構造情報がリレーションナルデータベース13に登録される。また、分解された部分構造は編集単位処理部8に入力され、各要素の情報が要素情報として、また部分構造中の各要素間の関係が構造情報として、リレーションナルデータベース13に登録される。2要素間の先祖・子孫関係は、グローバル構造情報による部分構造間の先祖・子孫関係と、先祖側の部分構造の構造情報による先祖・子孫関係の判断によって、簡単に、しかも高速に求めることができる。



**【特許請求の範囲】**

**【請求項 1】** 構造化文書を管理する構造化文書管理装置において、入力された構造化文書を設定に従って複数の部分構造に分解するとともに前記部分構造間の関係を第1の構造情報として生成する分解手段と、前記分解手段によって分解された部分構造ごとに該部分構造中の各要素間の関係を第2の構造情報として生成する構造情報登録手段と、前記分解手段で生成した第1の構造情報と前記構造情報登録手段で生成した第2の構造情報を保持する情報保持手段を有することを特徴とする構造化文書管理装置。

**【請求項 2】** 構造化文書を管理する構造化文書管理办法において、入力された構造化文書を設定に従って複数の部分構造に分解するとともに、前記部分構造間の関係を第1の構造情報として生成し、分解された部分構造ごとに該部分構造中の各要素間の関係を第2の構造情報として生成し、前記第1の構造情報と前記第2の構造情報を保持して前記構造化文書を管理することを特徴とする構造化文書管理方法。

**【請求項 3】** 構造化文書中の要素間の先祖・子孫関係を検索条件とした構造検索を行う検索装置において、構造化文書を設定に従って分解した複数の部分構造間の関係を示す第1の構造情報と前記部分構造ごとに該部分構造中の各要素間の関係を示す第2の構造情報を保持する情報保持手段と、それぞれの要素が含まれる部分構造間の先祖・子孫関係を前記第1の構造情報によって求めるとともに部分構造が先祖・子孫関係にあるときはさらに先祖側の部分構造から子孫側の部分構造へのパス上にあって前記先祖側の部分構造の子の部分構造のルートとなる要素と前記先祖側の部分構造に含まれる要素との間の先祖・子孫関係を前記第2の構造情報によって求める構造検索手段を有することを特徴とする検索装置。

**【請求項 4】** 前記構造検索手段は、前記先祖側の部分構造に含まれる要素が当該先祖側の部分構造のルートとなる要素である場合には、前記第2の構造情報を用いた判定を行わないことを特徴とする請求項3に記載の検索装置。

**【請求項 5】** 前記構造検索手段は、2つの要素が同一の部分構造に含まれているときには前記第2の構造情報を用いて先祖・子孫関係を求ることを特徴とする請求項3または請求項4に記載の検索装置。

**【請求項 6】** 構造化文書中の要素間の先祖・子孫関係を検索条件とした構造検索を行う検索方法において、構造化文書を設定に従って分解した複数の部分構造間の関係を示す第1の構造情報と前記部分構造ごとに該部分構造中の各要素間の関係を示す第2の構造情報を保持しておき、それぞれの要素が含まれる部分構造間の先祖・子孫関係を前記第1の構造情報によって求めるとともに、部分構造が先祖・子孫関係にあるときは、さらに先祖側の部分構造から子孫側の部分構造へのパス上にあって前

記先祖側の部分構造の子の部分構造のルートとなる要素と前記先祖側の部分構造に含まれる要素との間の先祖・子孫関係を前記第2の構造情報によって求めることを特徴とする検索方法。

**【請求項 7】** 前記先祖側の部分構造に含まれる要素が当該先祖側の部分構造のルートとなる要素である場合には、前記第2の構造情報を用いた判定を行わないことを特徴とする請求項6に記載の検索方法。

**【請求項 8】** 2つの要素が同一の部分構造に含まれているときには前記第2の構造情報を用いて先祖・子孫関係を求ることを特徴とする請求項6または請求項7に記載の検索方法。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】** 本発明は、大規模な構造化文書を編集可能に管理する構造化文書管理装置及び構造化文書管理办法と、管理されている構造化文書中の要素間の先祖・子孫関係を検索する検索装置及び検索方法に関するものである。

**【0002】**

**【従来の技術】** 従来より、XMLなどの構造化文書をデータベース化し、内容や文書構造での検索や、部分的な再利用などの利用に供している。一般に数ページ程度の小さな文書では、編集者が1人で文書の作成、編集を行う場合が多く、構造化文書をデータベース化した場合でも、文書全体を編集対象として編集作業が行われる場合が多い。しかし、大規模な文書では複数人が文書中のそれぞれの担当部分を編集してゆくといったことが行われており、構造化文書をデータベース化した場合でも、このような共同編集環境の提供が必要となる。

**【0003】** 従来、構造化文書をデータベース化する場合には、例えば文書中の各要素の親子関係を示すボインタなどで結んで文書中の構造を保持したり、構造を示すインデックスペーパルなどを用いて構造を保持している。他にもいくつかの方法が考えられているが、従来の方法はいずれも、文書全体を一律に画一的なデータ形式によって保持している。

**【0004】** このような従来の構造化文書の格納方法を用いている場合、編集した文書を格納すると、その影響が文書全体に波及してしまうことも少なくない。そのため、上述のような共同編集環境を考えた場合、従来の方法では複数人がそれぞれ並行して編集することができなかつた。

**【0005】** また、編集により文書全体に影響が及ぶ場合、文書に対して編集が加えられる度に、文書全体のデータを再構築する必要が発生する。大規模な文書では、このようなデータの再構築には多大な処理時間がかかり、編集するたびにデータの再構築を行っていたのでは非常に効率が悪いという問題があった。

**【0006】** 一方、このように構造化文書をデータベー

ス化した場合、文書構造の検索、特に2つの要素間の先祖・子孫関係の検索が頻繁に行われる。先祖・子孫関係は、ある要素から上位層へ要素をたぐってゆくだけで、あるいは下位層へ要素をたぐってゆくだけで、たどり着くことができる要素間の関係を示す。2つの要素が上下関係で直接接続されている関係が親子関係であるが、先祖・子孫関係は親子関係を含むものである。

【0007】このような先祖・子孫関係の検索は、頻繁に行われることから高速に実行できることが望まれる。従来は要素を1つずつたぐってゆくことによって行うが、枝分かれが多く、また階層が深い文書では、要素の探索時間も長くなってしまう。そのため、大規模な文書において先祖・子孫関係の検索を効率よく行うことができると手法が要望されていた。

#### 【0008】

【発明が解決しようとする課題】本発明は、上述した事情に鑑みてなされたもので、大規模な構造化文書を効率よく取り扱うことができるとともに、複数人が共同して編集可能な環境を提供できる構造化文書管理装置及び構造化文書管理方法を提供することを目的とするものである。さらに、このような構造化文書管理装置及び構造化文書管理方法により管理されている構造化文書中の要素の先祖・子孫関係を効率よく判定して文書構造の検索を高速に行うことができる検索装置及び検索方法を提供することを目的とするものである。

#### 【0009】

【課題を解決するための手段】本発明は、構造化文書を管理する構造化文書管理装置及び構造化文書管理方法において、入力された構造化文書を設定に従って複数の部分構造に分解するとともに、部分構造間の関係を第1の構造情報として生成する。さらに、分解された部分構造ごとに、部分構造中の各要素間の関係を第2の構造情報として生成する。そして、第1の構造情報と第2の構造情報を保持して、構造化文書を管理する。このように、構造化文書を部分構造に分解して、その構造を第1及び第2の構造情報として保持するので、部分構造単位での編集を行うことによって、編集による影響を部分構造内に収めることができる。従って、複数人がそれぞれ担当する部分構造を並行して編集する場合でも、他の部分構造への影響をなくし、共同編集環境を提供することができる。また、構造が変更された場合でも、そのためのデータの再構築は部分構造内で行えばよいので、高速に再構築のための処理を行うことができる。

【0010】また本発明は、このように構造化文書を設定に従って分解した複数の部分構造間の関係を示す第1の構造情報と前記部分構造ごとに該部分構造中の各要素間の関係を示す第2の構造情報を保持している場合に、構造化文書中の要素間の先祖・子孫関係を検索条件とした構造検索を行う検索装置及び検索方法であって、まず、それぞれの要素が含まれる部分構造間の先祖・子孫

関係を第1の構造情報によって求める。部分構造が先祖・子孫関係にあるときは、さらに先祖側の部分構造から子孫側の部分構造へのパス上にあって先祖側の部分構造の子の部分構造のルートとなる要素と、先祖側の部分構造に含まれる要素との間の先祖・子孫関係を、第2の構造情報によって求める。これによって、従来のようにそれぞれの要素を1つずつたぐってゆかなくても、2つの要素が先祖・子孫関係にあるか否かを高速に求めることができ、大規模な構造化文書の検索を効率的に行うことができる。

【0011】なお、この検索を行う際に、先祖側の部分構造に含まれる要素が当該先祖側の部分構造のルートとなる要素である場合には、前記第2の構造情報を用いた判定を行わず、第1の構造情報を用いた処理のみで2つの要素が先祖・子孫関係にあるか否かを求めることができる。また、2つの要素が同一の部分構造に含まれているときには、第1の構造情報を用いることなく、第2の構造情報を用いて先祖・子孫関係を求めることができる。

#### 【0012】

【発明の実施の形態】図1は、本発明の実施の一形態を示すブロック図である。図中、1は文書入力部、2はファイル管理部、3はディスパッチャ、4は検索部、5は文書処理部、6は文書解析部、7は文書分解部、8は編集単位処理部、9は文書解析部、10は差分解析部、11は要素情報登録部、12は構造情報登録部、13はリレーショナルデータベースである。文書入力部1は、構造化文書をファイル管理部2に対して入力する。このとき、ここでは分解前の構造化文書であるか、分解された部分構造であるかを示すコンテンツタイプも入力するものとする。また文書入力部1は、ファイル管理部2の検索部4に対して構造化文書あるいは構造化文書の部分構造の取り出しを要求して部分構造を受け取る。受け取った構造化文書あるいはその部分構造に対して編集を行った結果を再びファイル管理部2に対して入力することができる。また、構造化文書に対する文書構造の検索などを要求することができる。

【0013】ファイル管理部2は、文書入力部1との間で構造化文書あるいはその部分構造の受け渡しを行う。ファイル管理部2は、ディスパッチャ3及び検索部4を有している。ディスパッチャ3は、文書入力部1から渡される構造化文書あるいはその部分構造を受け取り、そのコンテンツタイプに従って、部分構造への分解前の構造化文書の場合には、その構造化文書を文書処理部5に渡す。また、分解された部分構造を受け取った場合には、その部分構造を編集単位処理部8へ渡す。さらに、文書処理部5で分解された部分構造を受け取って編集単位処理部8へ渡す。

【0014】検索部4は、文書入力部1から要求された構造化文書あるいはその部分構造をリレーショナルデータ

タベース 1 3 から取り出し、文書入力部 1 へ渡す。また、各種のリレーションナルデータベース 1 3 に対する検索要求を受け付け、検索結果を返す。特に文書構造の検索を行うことができ、このとき、本発明の検索方法による要素間の先祖・子孫関係の判定を行う。

【0015】文書処理部 5 は、ファイル管理部 2 のディスパッチャ 3 から渡される構造化文書を解析して、部分構造に分解する。文書処理部 5 は、構造化文書を解析する文書解析部 6 と、リレーションナルデータベース 1 3 から読み込んだ設定に従って構造化文書を複数の部分構造に分解する文書分解部 7 を有している。大規模な構造化文書では、分解された部分構造が編集単位となる。分解された部分構造には部分構造であることを示すコンテンツタイプが付加され、ファイル管理部 2 に入力される。また、部分構造間の関係をグローバル構造情報（第 1 の構造情報）として、リレーションナルデータベース 1 3 にレコードとして登録する。

【0016】編集単位処理部 8 は、ファイル管理部 2 から渡される部分構造について、部分構造内の要素の情報を要素情報として、また各要素間の関係を構造情報として、それぞれリレーションナルデータベース 1 3 に格納する。編集単位処理部 8 は、文書解析部 9、差分解析部 10、要素情報登録部 11、構造情報登録部 12 を有している。文書解析部 9 は、ファイル管理部 2 から渡された部分構造を解析する。

【0017】この例では、編集された部分構造については編集前の部分構造との差分をリレーションナルデータベース 1 3 に格納することにより版管理の利用に供するものとし、そのための構成として差分解析部 10 を有している。差分解析部 10 は、文書解析部 9 による解析結果に従い、リレーションナルデータベース 1 3 に保持されている編集前の部分構造と編集後の部分構造の差分を抽出してリレーションナルデータベース 1 3 への登録対象とする。なお、新規の構造化文書の部分構造である場合には、部分構造全体をリレーションナルデータベース 1 3 への登録対象とする。差分管理を行わない場合には、この差分解析部 9 は不要である。

【0018】要素情報登録部 11 は、部分構造中の各要素について、それぞれの属性情報や内容をフィールド値とするレコードをリレーションナルデータベース 1 3 に登録する。なお、差分解析部 10 による差分の抽出で追加されたことが判明した要素については新規の登録を行い、変更及び削除されたことが判明した要素については削除処理を行う。削除処理は、実際にレコードを削除してもよいし、例えばバージョン管理を行う場合には、削除された旨の情報を付加して残しておいてよい。

【0019】構造情報登録部 12 は、ファイル管理部 2 から渡された部分構造について、各要素間の関係を構造情報（第 2 の構造情報）として生成し、リレーションナルデータベース 1 3 に登録する。部分構造に対して編集が

行われた場合には、その部分構造の構造情報を再構築して新たに登録する。構造情報は、このように編集が行われると再構築が必要となるが、分解された部分構造ごとに行うので、他の部分構造には影響を与えない更新することが可能であり、文書全体の文書構造の再構築を行いう必要はない。なお、バージョン管理を行う際には、当該部分構造が編集される都度、構造情報を再構築してグローバル構造情報との対応付けを行っておけばよい。

【0020】リレーションナルデータベース 1 3 は、一般的なリレーションナルデータベースであり、構造化文書の各要素の要素情報と、部分構造間の関係を示すグローバル構造情報と、部分構造内の各要素間の関係を示す構造情報をそれぞれレコードとして格納し、これらの情報によって構造化文書を保持する。なお、グローバル構造情報及び構造情報については、その検索性能を高めるため、バイナリ形式のデータとしてレコード中のフィールドに格納しておくとよい。もちろんこれらのデータを別のデータベースに登録してもよい。

【0021】次に、上述の本発明の実施の一形態における概括的な動作の一例を説明する。図 2 は、構造化文書の構造の一例の説明図、図 3 は、グローバル構造情報及び要素情報と構造情報の一例の説明図である。ここでは図 2 (A) に示すように 5 階層の構造化文書が入力される場合を考える。各要素を○印で示しており、数字は各要素を特定するシーケンス番号 (S N o) である。

【0022】文書入力部 1 から図 2 (A) に示すような構造化文書（及びコンテンツタイプ）が入力されると、ディスパッチャ 3 は、入力された構造化文書のコンテンツタイプに応じ、ここでは文書処理部 5 へ入力された構造化文書を渡す。

【0023】文書処理部 5 では、文書解析部 6 で構造化文書を解析し、文書分解部 7 でリレーションナルデータベース 1 3 から読み込んだ設定に従い、文書を複数の部分構造に分解する。例えば図 2 (A) において三角形で示した部分ごとに構造化文書を分解する。これによって図 2 (B) に示すように 3 つの部分構造となる。分解された部分構造は、再びファイル管理部 2 のディスパッチャ 3 に渡される。

【0024】このように部分構造への分解を行ったときに、各部分構造の関係を示すグローバル構造情報を生成してリレーションナルデータベース 1 3 に格納する。図 2 (B) に示すように分解した 3 つの部分構造にそれぞれグローバル ID (G I D と略す) として 0, 1, 2 と振り、構造を示す情報として、説明を簡単にするため親の G I D を保持するとすれば、図 3 (A) に示すようなそれぞれの部分構造の親の G I D を並べたデータが得られる。このようなデータを、例えばメモリイメージとしてそのままレコード中のフィールドとして埋め込み、ここでは文書を一意に特定するための文書 I D とともにリレーションナルデータベース 1 3 に格納する。

【0025】なお、部分構造へ分解する際に、下位の部分構造の頂点の要素が、その上位の部分構造にも含まれることになる。例えば図2に示した例では、SNoが3、8の要素については複数の部分構造にまたがる。このような場合には、上位の部分構造にダミーの要素を附加しておく。この例ではGIDが0の部分構造に、SNoが3の要素の代わりにSNoが13の要素を、SNoが8の要素の代わりにSNoが14の要素を仮想的に附加している。なお、附加したダミーの要素を、マウントポイントと呼ぶことにする。

【0026】文書分解部7で分解された部分構造を受け取ったディスパッチャ3では、今度は部分構造を編集単位処理部8に渡す。編集単位処理部8では、文書解析部9で部分構造を解析し、差分解析部10に渡す。ここでは新規の構造化文書であるので、差分解析部10は部分構造を要素情報登録部11及び構造情報登録部12に渡す。

【0027】要素情報登録部11では、部分構造中のそれぞれの要素に関する情報を要素情報としてリレーションナルデータベース13に登録する。例えば図2(B)に示したGIDが2の部分構造の場合、SNoが8~12の各要素の情報をリレーションナルデータベース13に登録することになる。図3(B)に要素情報の一例を示しており、この例では各要素を特定するSNoとともに、部分構造中で要素を特定するローカルナンバ(LNo)、要素名、属性値をフィールド値としている。もちろん、フィールドの構成は任意であり、このほかにバージョン情報などを含んでいてもよい。この要素情報では各要素間の親子関係などは含まれていないが、例えば属性値の検索など、フィールド値による高速な検索が可能である。また、後述する構造情報とは、例えばLNoやSNoによって対応づけることができる。

【0028】構造情報登録部12では、部分構造中の各要素間の関係を示す構造情報を生成する。構造情報は、どのようなデータ構造であってもよいが、図3(C)に示す例では、それぞれの要素の親のLNoを並べたデータとして構成している。また、この例では構造情報のメモリイメージをそのままレコード中のフィールドとして埋め込み、さらに部分構造を一意に特定するためのGIDとともにリレーションナルデータベース13に格納する。

【0029】なお、図3(B)、(C)には、GIDが2の部分構造における要素情報及び構造情報のみを示しているが、同様の要素情報及び構造情報が、GIDが0、1の部分構造についても作成され、リレーションナルデータベース13に登録される。また、図3に示したグローバル構造情報、要素情報、構造情報のデータ構造は一例であり、リレーションナルデータベース13の設計時に任意に取り決めておくことができる。例えば後述する具体例においては、図3に示したデータ構造とは一部異

なるデータ構造を用いている。

【0030】上述のようにして構造化文書が分解されてリレーションナルデータベース13に登録されていると、部分構造を編集単位として文書の編集を行うことができる。例えば文書入力部1から部分構造の取り出しをファイル管理部2に要求することによって、検索部4が編集すべき部分構造をリレーションナルデータベース13から読み出して文書入力部1に対して出力する。そして、文書入力部1において部分構造に対して編集処理を行った後、再びファイル管理部2に入力する。

【0031】ファイル管理部2に入力された編集後の部分構造(及びコンテンツタイプ)は、ディスパッチャ3によって編集単位処理部8へ送られる。編集単位処理部8では、文書解析部9で部分構造を解析し、差分解析部10に渡す。差分解析部10では、リレーションナルデータベース13に登録されている編集前の部分構造と文書解析部9から受け取った編集後の部分構造を比較し、その差分を抽出する。要素の編集としては、内容や属性値の変更の他、新たな追加、あるいは要素の削除などがある。要素情報登録部11では、新たに追加された要素については、要素情報をリレーションナルデータベース13に登録する。このとき、追加時のバージョン情報を付加してもよい。また、削除された要素については、レコードを削除したり、あるいは、当該要素に対応する要素情報に削除された旨の情報(例えば旧バージョン情報など)を付加する。変更の場合には、編集前の要素情報を削除、編集後の要素情報を追加する処理を行えばよい。

【0032】さらに、編集された部分構造に対応する構造情報の再構築を構造情報登録部12において行う。この場合、再構築する構造情報は部分構造の範囲内だけによく、構造化文書全体の構造情報を再構築する場合に比べて格段に少ない処理で済む。また、他の部分構造に影響しないので、他の部分構造を他人が編集中であっても全く問題なく再構築の処理を行うことができる。

【0033】また、文書入力部1からファイル管理部2に対して、文書構造の検索を要求することができる。要求を受け取った検索部4では、リレーションナルデータベース13に登録されている構造化文書の中から検索条件に適合した文書構造を有する文書を検索して、文書入力部1に検索結果を返す。この文書構造の検索の際には、多くの場合、2つの要素間の先祖・子孫関係の検索が行われる。上述のように構造化文書を部分構造に分解して、部分構造間の関係を示すグローバル構造情報と、各部分構造内の要素間の関係を示す構造情報とを有していることによって、階層や枝の多い、大規模な構造化文書でも、容易に検索を行うことができる。

【0034】まず、同じ部分構造内に存在する要素間の先祖・子孫関係の検索は、当該要素が存在する部分構造の構造情報を参照すればよい。部分構造であるので、要素数も少なく、高速に先祖・子孫関係の検索を行うこと

ができる。なお、部分構造内の先祖・子孫関係の検索方法は任意であり、従来から用いられている各種の文書構造の検索手法を用いることができる。

【0035】関係を検索する2つの要素が異なる部分構造に存在している場合には、まず、それぞれの要素が存在する部分構造間の先祖・子孫関係の検索を行う。この検索には、グローバル構造情報を参照すればよい。部分構造の数は要素の数に比べて格段に少なく、この検索は高速に実行可能である。またこの場合の検索手法は、それぞれの部分構造を1つの要素と考えれば従来から用いられている各種の文書構造の検索手法を用いることができる。

【0036】それぞれの要素が存在する部分構造が先祖・子孫の関係にある場合には、さらに、先祖側の部分構造から子孫側の部分構造へのパス上にあって、先祖側の部分構造の子の部分構造のルートとなる要素（すなわちマウントポイント）と先祖側の部分構造に含まれる要素との間の先祖・子孫関係を、先祖側の部分構造の構造情報によって求める。

【0037】例えば図2（A）に示した例において、SN<sub>o</sub>=6の要素とSN<sub>o</sub>=9の要素との先祖・子孫関係を求める場合、SN<sub>o</sub>=6の要素が含まれるGID=0の部分構造と、SN<sub>o</sub>=9の要素が含まれるGID=2の部分構造とは先祖・子孫の関係にあり、さらに、GID=2の部分構造のルートとなる要素（SN<sub>o</sub>=8でありSN<sub>o</sub>=14もある）とSN<sub>o</sub>=6との先祖・子孫関係を、GID=0の部分構造の構造情報を用いて調べる。これは先祖・子孫の関係にあるため、結果として、SN<sub>o</sub>=6とSN<sub>o</sub>=9とは先祖・子孫関係にあると判断される。

【0038】同じGID=0の部分構造の要素でも、SN<sub>o</sub>=2の要素の場合、部分構造間の関係は先祖・子孫関係となるものの、SN<sub>o</sub>=2とSN<sub>o</sub>=14との関係は先祖・子孫関係にないため、SN<sub>o</sub>=2とSN<sub>o</sub>=9の要素は先祖・子孫関係ないと判断される。

【0039】このように、部分構造間の先祖・子孫関係を調べた後に、先祖側の部分構造内の関係を調べるだけで、2つの要素間の先祖・子孫関係を調べることができる。要素数が膨大な、大規模な文書においては、従来のような要素を1つずつたぐってゆく検索手法では、このような先祖・子孫関係の検索は非常に時間のかかる処理である。しかし本発明によれば、短時間で、簡単に、先祖・子孫関係の検索を行うことができる。

【0040】なお、先祖側の部分構造に含まれる要素が、その部分構造のルートとなる要素である場合には、部分構造間の先祖・子孫関係の検索を行うだけでよい。なぜなら、その部分構造のルートとなる要素であれば、その部分構造の下位に存在するいずれの部分構造内の要素の親であることが明白だからである。このように後続の判定を行わないことによって、検索処理をさらに高速

化することができる。

【0041】以下、具体例を用いながら上述の動作についてさらに詳細に説明してゆく。図4は、具体例におけるデータ構造の一例の説明図である。以下に説明する具体例では、図4に示すようなデータ構造を用いてグローバル構造情報、要素情報、構造情報をリレーションナルデータベース13に登録し、利用する。

【0042】まずグローバル構造情報は、図4（A）に示すように、それぞれの部分構造ごとに、最大GID、親GID、親の接続ID、編集単位IDなどの情報を有し、これらの部分構造ごとのデータをGIDの順に並べて構成されている。GIDは、ルートノードから深さ優先、左優先で探索される部分構造に対して順に振られたIDである。このGIDの振り方は、後述する部分構造内の構造情報において各要素に振るノードオーダーと同様であり、後述するノードオーダーの振り方の説明によりGIDの振り方の説明に代える。最大GIDは、当該部分構造よりも下位の部分構造のGIDのうち最大のGIDである。深さ優先でGIDを振っているので、当該部分構造の下位の部分構造には、当該部分構造のGIDから最大GIDまでのGIDが振られている。また、最大GIDよりも大きいGIDは、当該部分構造の下位には存在しないことが保証されている。親GIDは、当該部分構造の親となる部分構造のGIDである。親の接続IDは、例えば図2（B）に示す例ではGIDが1の部分構造は、GIDが0の部分構造において仮想的に付加した要素のうちのいずれに接続されるかを示すものである。編集単位IDは、当該部分構造をシステム全体で一意に識別するためのIDである。これらの情報は、例えばGIDの数だけ並べてバイナリ形式でレコード中の1つのフィールド値とし、そのほかに文書IDなどのフィールドを付加して、リレーションナルデータベース13に保持させることができる。もちろん、保存時のデータ形式は任意である。

【0043】要素情報は、図4（B）に示すように、ノードテーブル、属性テーブル、テキストテーブルによって構成される。要素のうち、リーフとなる要素についてはテキストテーブルの情報を持ち、その他の要素についてはノードテーブル及び属性テーブルの情報をを持つことになる。例えばノードテーブルと属性テーブルを一つのテーブルにするなど、変形は任意である。

【0044】ノードテーブルは、上述のSN<sub>o</sub>、LN<sub>o</sub>、編集単位ID、要素名、文書ID、追加バージョン情報、削除バージョン情報、マウントポイント数などのフィールドを有している。SN<sub>o</sub>はシステム全体において一意に要素に振られたIDである。LN<sub>o</sub>は、部分構造内で一意に要素に振られたIDである。このLN<sub>o</sub>は、文書構造とは無関係である。編集単位IDは、当該要素を含んでいる部分構造を特定するIDである。要素名は、文書中の要素の名前である。文書IDは、当該要

素を含んでいる文書を特定する ID である。追加バージョン情報は、当該要素が追加されたバージョンを示す情報である。新規に文書が登録された時点では 1 となる。削除バージョン情報は、当該要素が削除される直前のバージョンを示す情報である。新規に文書が登録された時点では NULL であり、例えば最初の更新で削除されれば 1 となる。マウントポイント数は、当該要素の下位

(子孫) に接続されるマウントポイントの数を示す。マウントポイントは、上述のように複数の部分構造にまたがる要素について、位の部分構造に付加したダミーの要素である。例えば図 2 (B) に示す G ID が 0 の部分構造では、S No が 1 の要素のマウントポイント数は 2 となる。また属性テーブルは、S No、属性名、属性値などのフィールドを有している。S No はノードテーブルと同様である。属性名は当該要素の属性に付された名前であり、その値が属性値として格納される。

【0045】テキストテーブルは、S No、L No、編集単位 ID、要素値、文書 ID、追加バージョン情報、削除バージョン情報などのフィールドを有している。要素値には、リーフとなる要素が有している文書内容が格納される。また、例えば図 2 (B) に示す例において G ID が 0 の部分構造に仮想的に付加した要素は、その部分構造においてはリーフとなる要素である。この場合には、要素値として接続 ID が格納される。他のフィールドについては上述の通りである。

【0046】構造情報は、図 4 (C) に示すように、部分構造内の各要素ごとに L No、ノードオーダー、最大ノードオーダーなどのデータが並べられている。L No は上述の通りである。ノードオーダーは、部分構造内のルートとなる要素から深さ優先、左優先で探索される要素に対して順に振られた ID である。最大ノードオーダーは、部分構造中で当該要素よりも下位の要素のノードオーダーのうち最大のノードオーダーである。深さ優先でノードオーダーを振っているので、当該要素の下位の要素には、当該要素のノードオーダーから最大ノードオーダーまでの ID が振られている。また、最大ノードオーダーよりも大きい ID は、当該要素の下位には存在しないことが保証されている。これらの情報は、例えば、部分構造内に存在する要素の数(仮想的に設けた要素も含めて)だけ並べてバイナリ形式でレコード中の 1 つのフィールド値とし、そのほかにクラスタ ID などのフィールドを付加してリレーションデータベース 13 に保持させることができる。もちろん、保存時のデータ形式は任意である。なお、クラスタ ID は、上述の編集単位 ID と同様に部分構造を特定する ID であるが、構造情報は文書の編集などにより版が変わったときに再構築されるので、版が変わることに新たなクラスタ ID が付与される。バージョン(版)管理を行わない場合には、クラスタ ID は編集単位 ID でよい。

【0047】バージョンテーブルは、図 4 (D) に示す

ように、クラスタ ID、編集単位 ID、バージョン情報、文書 ID などによって構成されている。このバージョンテーブルによって、グローバル構造情報において部分構造に付した編集単位 ID と、バージョンごとに異なる部分構造内の文書構造を示す構造情報を結びつけている。なお、別途、各文書について最新のバージョンを示す情報が保持されており、最新の文書を取り出す際にはその最新バージョン情報からバージョンテーブルを検索することによって、各部分構造についての最新の構造情報を得ることができる。また、属性情報については、上述のように削除バージョン情報が NULL のレコードを取り出せば、最新の要素情報を取り出すことができる。バージョン(版)管理を行わない場合には、このバージョンテーブルを設げずに構成することもできる。

【0048】図 5 は、入力される構造化文書の具体例の説明図である。ここでは、XML によって記述された、図 5 (A) に示すような構造化文書が文書入力部 1 から入力されるものとする。この構造化文書の文書構造をツリー表現すると図 5 (B) に示すようになる。なお、図 5 (B) において○は要素を示し、内部の数字は S No を示している。抜けている S No については、後述するようにマウントポイントに割り当てられる。

【0049】このような構造化文書が文書入力部 1 から入力されると、ディスパッチャ 3 は、入力された構造化文書を文書処理部 5 へ渡す。文書処理部 5 では、文書解析部 6 で構造化文書を解析し、文書分解部 7 でリレーションデータベース 13 から読み込んだ設定に従い、文書を複数の部分構造に分解する。ここでは CHAPTER、SECTION で分解することとする。

【0050】図 6～図 8 は、分解された部分構造ごとの文書の具体例の説明図である。各図において、分解された文書の記述を (A) として示し、分解された部分構造のツリー表現を (B) として示している。以下の説明では図 6 に示す部分構造を編集単位 ID = 0、図 7 に示す部分構造を編集単位 ID = 1、図 8 に示す部分構造を編集単位 ID = 2 とする。また、この編集単位 ID とは別に、各部分構造には G ID が振られる。ここでは、図 6 に示す部分構造に G ID = 0、図 7 に示す部分構造に G ID = 1、図 8 に示す部分構造に G ID = 2 を振るものとする。

【0051】図 6 に示す部分構造では、部分構造への分解の際に、図 5 (B) に示す S No = 6 の要素に対応する仮想的な要素 (S No = 16) を付加している。図 6 (A) に示す文書の記述では、分解により他の部分構造に移した記述の部分を “& E 001 ;” という記述に置き換えている。これが接続 ID となる。図 7 に示す部分構造についても同様であり、図 5 (B) に示す S No = 9 の要素に対応する仮想的な要素 (S No = 18) を付加している。図 7 (A) に示す文書の記述では、分解により他の部分構造に移した記述の部分を “& E 00

2 ;”という接続IDに置き換えている。この置き換えられた要素がマウントポイントとなる。

【0052】図9は、グローバル構造情報の具体例の説明図である。上述のように構造化文書を部分構造に分解すると、その時点で各部分構造間の関係がわかる。各部分構造間のツリー表現を図9（B）に示している。図9（B）において、各部分構造を三角形で表している。三角形の中には、GIDとともにマウントポイントの接続IDを示している。

【0053】この部分構造間の関係をグローバル構造情報としてリレーショナルデータベース13に登録する。この具体例では、グローバル構造情報は例えば図9（A）に示すようになる。なお、GID=1の部分構造は、GID=0の部分構造中のマウントポイント（SN<sub>o</sub>=16）に接続されるので、その接続ID（“&E001；”）のうちの数値部分のみを親の接続IDとしてグローバル構造情報中に含めている。GID=2の部分構造についても同様であり、親の接続IDとしてGID=1の部分構造中のマウントポイント（SN<sub>o</sub>=18）の接続ID（“&E002；”）のうちの数値部分のみを親の接続IDとしてグローバル構造情報中に含めている。

【0054】このようなグローバル構造情報は、文書IDを付加してリレーショナルデータベース13に登録する。このとき、例えばバイナリ形式のままフィールド値として格納しておくことによって、検索時のパフォーマンスを向上させることができる。

【0055】図6（A）、図7（A）、図8（A）に示した、分解された部分構造の記述は、再びファイル管理部2に戻される。ファイル管理部2のディスパッチャ3は、これらの分解された部分構造の記述を編集単位処理部8に渡す。編集単位処理部8では、文書解析部9で各部分構造の記述を解析し、差分解析部10に渡す。差分解析部10では、新規の構造化文書であるので、文書解析部9による部分構造の解析結果を要素情報登録部11及び構造情報登録部12に渡す。

【0056】要素情報登録部11では、各部分構造の記述から、要素情報を生成する。図4（B）に示した要素情報のデータ構造に従い、リーフに対応する要素についてはテキストテーブルを生成し、それ以外の要素についてはノードテーブル及び属性テーブルを生成する。図10、図11は、要素情報の具体例の説明図である。図10（A）はノードテーブル、図10（B）は属性テーブル、図11はテキストテーブルを示している。SN<sub>o</sub>は、図6～図8では説明のためにすでに振られているが、実際には編集単位処理部8に各部分構造の記述が入力されて解析することによって、要素の出現順に振られる。LN<sub>o</sub>についても、各部分構造内で文書構造には関係なく振られる。追加バージョンには、新規登録時には1が格納される。また削除バージョンには、新規登録時

には‘NULL’が格納される。マウントポイント数には、仮想的な要素（SN<sub>o</sub>=12, 14）が下位に接続されているSN<sub>o</sub>=0, 3, 6の要素について‘1’となり、他の要素については0となる。このような要素情報がリレーショナルデータベース13に登録される。

【0057】構造情報登録部12では、部分構造中の各要素間の関係を示す構造情報を生成する。図4（C）に示した構造情報のデータ構造に従い、それぞれの要素のLN<sub>o</sub>、ノードオーダー、最大ノードオーダーを求めて格納してゆく。図12は、構造情報を生成する処理の一例を示すフローチャートである。ここではXMLのタグ及びテキストが順次切り出され、そのたびにそれぞれの処理を行うものとし、またLN<sub>o</sub>についても別途付与されるものとして説明する。なお、処理を進めるためにLN<sub>o</sub>及びインデックスの値を格納するためのスタックを用いる。

【0058】S21において、初期設定としてインデックスを0にセットしておく。S22において、タグまたはテキストの切出を検知し、LN<sub>o</sub>を取得する。S23において、タグまたはテキストが存在していたか否かを判定し、タグまたはテキストを切り出すことができず、部分構造の記述が終了していればこの処理を終了する。

【0059】S24において、切り出された要素がテキストか否かを判断し、切り出された要素がタグであればさらにS25において開始タグか終了タグかを判定する。切り出された要素が開始タグの場合には、S26において、S22で取得したLN<sub>o</sub>と現在のインデックスをスタックに積む。そしてS27において、インデックスの値を1だけ増加させる。

【0060】切り出された要素がテキストの場合には、S28において、現在のインデックスの値をノードオーダー及び最大ノードオーダーとし、S22で取得したLN<sub>o</sub>に対応する構造情報の位置に、LN<sub>o</sub>、ノードオーダー、最大ノードオーダーを格納する。

【0061】切り出された要素が終了タグの場合には、S29において、スタックからLN<sub>o</sub>及びインデックスを取り出し、S30において、ノードオーダーとしてスタックから取り出したインデックスの値、最大ノードオーダーとして現在のインデックスの値をセットし、スタックから取り出したLN<sub>o</sub>に対応する構造情報の位置に、LN<sub>o</sub>、ノードオーダー、最大ノードオーダーを格納する。

【0062】このような処理を行うと、ノードオーダーとして、それぞれの要素には深さ優先、左優先で探索された順に番号付けが行われる。図13は、編集単位IDが2の部分構造における構造情報生成過程の一例の説明図である。図10（A）及び図11に示したように、各要素にはLN<sub>o</sub>が振られている。このLN<sub>o</sub>は文書構造とは関係なく振られている。図8（A）に示すような編集単位IDが2のXMLによる記述をもとに、図12に

示した処理を実行する。なお、図13において矩形で囲んだ3つ組の数値は、左からLN<sub>o</sub>、ノードオーダー、最大ノードオーダーである。

【0063】LN<sub>o</sub>=0, 1, 2についてはLN<sub>o</sub>とそのときのインデックス(0, 1, 2)がスタックに積まれる。LN<sub>o</sub>=5のリーフノードの要素において、LN<sub>o</sub>=5、ノードオーダー=3、最大ノードオーダー=3が構造情報として格納される。スタックからLN<sub>o</sub>=2とインデックス=2が取り出され、LN<sub>o</sub>=2、ノードオーダー=2、最大ノードオーダー=3が構造情報として格納される。さらにスタックからLN<sub>o</sub>=1とインデックス=1が取り出され、LN<sub>o</sub>=1、ノードオーダー=1、最大ノードオーダー=3が構造情報として格納される。

【0064】次にLN<sub>o</sub>=3, 4についてはスタックに当該LN<sub>o</sub>とインデックス=4, 5が積まれる。LN<sub>o</sub>=6のリーフノードの要素において、LN<sub>o</sub>=6、ノードオーダー=6、最大ノードオーダー=6が構造情報として格納される。スタックからLN<sub>o</sub>=4とインデックス=5が取り出され、LN<sub>o</sub>=4、ノードオーダー=5、最大ノードオーダー=6が構造情報として格納される。さらにスタックからLN<sub>o</sub>=3とインデックス=4が取り出され、LN<sub>o</sub>=3、ノードオーダー=4、最大ノードオーダー=6が構造情報として格納される。さらにまた、スタックからLN<sub>o</sub>=0とインデックス=0が取り出され、LN<sub>o</sub>=0、ノードオーダー=0、最大ノードオーダー=6が構造情報として格納される。

【0065】図14は、構造情報及びバージョンテーブルの具体例の説明図である。上述のような処理によって、各部分構造ごとに図14(A)～(C)に示すような構造情報が得られる。特に編集単位IDが2の部分構造においては、図13を用いて詳述したような処理によって、図14(C)に示すような構造情報が得られることになる。図14では構造情報をテーブル形式で表現しているが、これらのデータはバイナリ形式でレコード中のフィールドに格納され、さらにクラスタIDが付されてリレーションナルデータベース13に登録される。また、図14(D)に示すように、バージョンごとにクラスタIDと編集単位IDとを対応づけるバージョン情報のレコードが各部分構造ごとに生成され、リレーションナルデータベース13に登録される。

【0066】以上説明したようにして、図5に示した構造化文書は、図6～図8に示したように部分構造に分割され、部分構造の間の関係を示すグローバル構造情報と、各要素の情報を格納した要素情報と、部分構造内の各要素間の関係を示す構造情報、それに編集時に備えてバージョンテーブルが生成され、リレーションナルデータベース13に登録される。

【0067】なお、文書の編集は部分構造ごとに行われる。編集後の部分構造はファイル管理部2に入力され、

ディスパッチャ3が編集単位処理部8へ渡す。編集単位処理部8では、渡された編集後の部分構造について、文書解析部9で解析し、差分解析部10は解析結果をもとにリレーションナルデータベース13に登録されている編集前の部分構造との差分を抽出する。そして、追加あるいは編集されて変更された後の要素については新たなバージョン情報を追加バージョン情報とした要素情報を生成し、リレーションナルデータベース13に登録する。また削除あるいは編集されて変更される前の要素については、編集直前のバージョン情報を削除バージョン情報として当該フィールドに書き込んでおく。また、編集された部分構造の構造情報を再構築し、新たなクラスタIDを付してリレーションナルデータベース13に登録し、また、対応するバージョンテーブルのレコードを生成して同じくリレーションナルデータベース13に登録する。

【0068】このようにして編集が加えられた場合のバージョン管理を行うことができる。このようなバージョン管理によって、例えば最新版の文書の要素情報を削除バージョンがNULLのレコードを取り出せばよく、簡単に最新版の文書あるいは部分構造を取り出すことができる。もちろん、このようなバージョン管理を行わないで構成することも可能である。その場合には、データ構造をより簡略化することができる。

【0069】次に、文書構造の検索を行う場合について具体例を用いて説明する。文書構造の検索を行う場合、その検索条件としては、ある属性を有する要素の下位に、ある属性を有する要素が存在する文書構造、といった条件が設定される場合が考えられる。ここではどのような検索条件による文書構造の検索を考え、一例として、要素の属性条件aを満たす要素の集合A{a<sub>i</sub> | 1 ≤ i ≤ N<sub>a</sub>}（ただしN<sub>a</sub>は属性条件aを満たす要素数。以下{a<sub>i</sub>}と記す。）に含まれる1つの要素が、要素の内容条件bを満たす要素の集合B{b<sub>j</sub> | 1 ≤ j ≤ N<sub>b</sub>}（ただしN<sub>b</sub>は属性条件bを満たす要素数。以下{b<sub>j</sub>}と記す。）のうち、いずれか一つ以上の要素を子孫を持つ集合を求める場合の動作について説明する。なお、要素の属性条件aを満たす要素の集合A{a<sub>i</sub>}、及び、要素の内容条件bを満たす要素の集合B{b<sub>j</sub>}は、例えば図10(A)に示すノードテーブル及び図11に示すテキストテーブルなどの要素情報に対して、属性値のフィールドで検索を行うことによって容易に求めることができる。

【0070】図15は、本発明の実施の一形態における2要素間の先祖・子孫関係の検索処理の一例を示すフローチャートである。ここでは集合A{a<sub>i</sub>}に含まれる1つの要素a<sub>i</sub>と、集合B{b<sub>j</sub>}に含まれる1つの要素b<sub>j</sub>との間の先祖・子孫関係を判定する処理を示している。なお、要素a<sub>i</sub>が含まれる部分構造をC<sub>a</sub><sub>i</sub>、要素b<sub>j</sub>が含まれる部分構造をC<sub>b</sub><sub>j</sub>とする。また、要素a<sub>i</sub>については、その要素a<sub>i</sub>が含まれる部分構造C<sub>a</sub>

$i$ において、要素 $a_i$ がルートとなる要素であるか否かが、わかっているものとする。

【0071】まずS41において、要素 $a_i$ と要素 $b_j$ が同じ文書に含まれているか否かを判定する。この判定は、要素情報中の文書IDを比較すればよい。別の文書であれば以降の処理を行わずに、先祖・子孫関係にはないと判定する。なお、例えば要素 $b_j$ を文書IDごとに昇順にソートしておくと、要素 $a_i$ の文書IDを上回った時点で以後の要素 $b_j$ については先祖・子孫関係にはないと判定して処理を終了することができる。もちろん、降順にソートしておいて、要素 $a_i$ の文書IDを下回った時点でも同様である。

【0072】要素 $a_i$ と要素 $b_j$ が同じ文書に含まれている場合、まずS42において、要素 $a_i$ が含まれる部分構造 $C_{a_i}$ と、要素 $b_j$ が含まれる部分構造 $C_{b_j}$ が同一の部分構造か否かを判定する。この判定は、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ のクラスタIDが一致するか否かを判定することにより行うことができる。部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ が同一である場合には、要素 $a_i$ と要素 $b_j$ は同じ部分構造内の要素である。この場合、当該部分構造内で要素 $a_i$ と要素 $b_j$ の先祖・子孫関係を判定すればよい。

【0073】このとき、S43において、要素 $a_i$ が部分構造 $C_{a_i}$ のルートノードであるか否かを判定する。要素 $a_i$ がルートノードである場合には、同じ部分構造内の要素 $b_j$ は必ず要素 $a_i$ の子孫である。従って、要素 $a_i$ と要素 $b_j$ は先祖・子孫関係にあると判定する。

【0074】要素 $a_i$ が部分構造 $C_{a_i}$ のルートノードではない場合には、S44において、要素 $a_i$ と要素 $b_j$ の先祖・子孫関係を判定する。この場合には要素 $a_i$ と要素 $b_j$ が同じ部分構造内であるので、当該部分構造の構造情報を用いて先祖・子孫関係を判定することができる。判定は、要素 $a_i$ のノードオーダー及び最大ノードオーダーと、要素 $b_j$ のノードオーダーを用い、(要素 $a_i$ のノードオーダー) < (要素 $b_j$ のノードオーダー)  $\leq$  (要素 $a_i$ の最大ノードオーダー) が成り立てば、要素 $b_j$ は要素 $a_i$ の子孫であると判定できる。この条件が満たされない場合、要素 $b_j$ は要素 $a_i$ の子孫ではないと判定できる。

【0075】要素 $a_i$ が含まれる部分構造 $C_{a_i}$ と、要素 $b_j$ が含まれる部分構造 $C_{b_j}$ が異なる部分構造である場合には、基本的には部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ の先祖・子孫関係を調べ、その後、要素 $a_i$ と部分構造 $C_{a_i}$ 内のマウントポイントとの関係を調べる。まずS45において、要素 $a_i$ のマウントポイント数を調べる。マウントポイント数が0であれば、下位の部分構造には要素 $a_i$ の子孫が存在しないことが明らかであるので、この時点で要素 $a_i$ と要素 $b_j$ は先祖・子孫関係にはないと判定できる。この判定によって、処理の高速化を図っている。またこの場合には、部分構造 $C_{a_i}$ に含

まれない要素 $b_j$ については、すべて先祖・子孫関係ではない。従って、要素 $a_i$ に対する検索においては、集合Bのうち、部分構造 $C_{a_i}$ に含まれない要素 $b_j$ については検索不要としてフラグを立てておくとよい。

【0076】要素 $a_i$ のマウントポイント数が0でないときには、S46において、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ との先祖・子孫関係を調べる。グローバル構造情報参照して部分構造 $C_{a_i}$ のGID、最大GIDと、部分構造 $C_{b_j}$ のGIDを取得し、

$(C_{a_i}\text{のGID}) < (C_{b_j}\text{のGID}) \leq (C_{a_i}\text{の最大GID})$

を判定する。この条件が成り立つとき、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ とは先祖・子孫関係にあると言える。この条件が成り立たないとき、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ とは先祖・子孫関係ではなく、従って要素 $a_i$ と要素 $b_j$ も先祖・子孫関係ではないと判定する。なお、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ が先祖・子孫関係でない場合には、部分構造 $C_{a_i}$ に含まれるすべての集合Aの要素について、部分構造 $C_{b_j}$ に含まれる集合Bの検索は不要であるというフラグを立ておくとよい。これによって不要な検索処理を行わずに済み、処理の高速化を図ることができる。

【0077】部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ が先祖・子孫関係にある場合には、次に要素 $a_i$ と部分構造 $C_{a_i}$ 内のマウントポイントとの関係を調べるが、処理量を軽減するために、S47において、要素 $a_i$ が部分構造 $C_{a_i}$ のルートノードであるか否かを判定する。要素 $a_i$ がルートノードである場合には、要素 $a_i$ は、部分構造 $C_{a_i}$ の下位の部分構造に含まれるすべての要素の先祖であることが必ず言える。従って、部分構造 $C_{a_i}$ と部分構造 $C_{b_j}$ が先祖・子孫関係であり、かつ、要素 $a_i$ が部分構造 $C_{a_i}$ のルートノードである場合には、要素 $a_i$ と要素 $b_j$ は先祖・子孫関係にあると判定する。

【0078】S48では、要素 $a_i$ と部分構造 $C_{a_i}$ 内のマウントポイントとの関係を調べる。部分構造 $C_{b_j}$ から部分構造 $C_{a_i}$ までのパスをたどり、そのパス上にあって、部分構造 $C_{a_i}$ の直下の部分構造のルートノードに対応する部分構造 $C_{a_i}$ におけるマウントポイントを求め、そのマウントポイントのノードオーダー(mとする)を取得する。この処理は、グローバル構造情報を用いて行うことができ、部分構造 $C_{b_j}$ から親GIDをたどってゆき、親GIDが部分構造 $C_{a_i}$ のGIDとなつたとき、その部分構造が部分構造 $C_{a_i}$ の直下の部分構造である。そして、その部分構造 $C_{a_i}$ の直下の部分構造のグローバル構造情報中の親の接続IDの値を取得することによって、部分構造 $C_{a_i}$ 中のマウントポイントを求めることができる。さらにマウントポイントの要素情報及び構造情報からノードオーダーを取得することができる。

【0079】そして、要素 $a_i$ のノードオーダーと最大ノードオーダー、それにマウントポイントのノードオーダー $m$ の関係が

(要素 $a_i$ のノードオーダー)  $\leq m \leq$  (要素 $a_i$ の最大ノードオーダー)

を満たすか否かを判定する。この条件を満たすとき、要素 $a_i$ とマウントポイントとは先祖・子孫関係にあり、よって、要素 $a_i$ と要素 $b_j$ とは先祖・子孫関係にあると判断することができる。また、この条件が満たされないとき、要素 $a_i$ とマウントポイントとは先祖・子孫の関係ではなく、よって、要素 $a_i$ と要素 $b_j$ とは先祖・子孫関係にはないと判断することができる。さらに、この条件が満たされないときには、要素 $a_i$ の検索において、部分構造 $C_{b_j}$ に含まれる集合Bの要素は検索しなくてよいものとしてフラグを立てておくことによって、検索処理を高速化することができる。

【0080】上述の処理はある要素 $a_i$ と要素 $b_j$ が特定された場合の先祖・子孫関係の判定処理であるが、このような処理を、集合A中のある要素 $a_i$ について、集合B中のそれぞれの要素 $b_j$ について行うことになる。要素 $b_j$ の選択の際には、S45, S46, S48において検索不要のフラグを立てた要素については検索対象から外すことによって、検索処理を高速化できる。

【0081】さらに、集合A中の要素を変更し、同様にして集合B内のそれぞれの要素との判定を行ってゆけばよい。このときにも、S46におけるフラグを考慮することによって、検索処理を高速化することができる。

【0082】検索結果としては、例えば検索条件が、ある属性を有する要素の下位に、ある属性を有する要素が存在する文書構造の存在するか否かのみである場合には、存在する、あるいは存在しないのみを結果とすればよい。この場合には、ある2つの要素間で先祖・子孫関係が確認できた時点で処理を終了することができる。また、ある属性を有する要素の下位に、ある属性を有する要素が存在する文書構造の実体の検索であれば、例えば条件を満たす集合Aの要素、あるいは要素を含む部分構造、または要素を含む文書ID、などを検索結果とすればよい。

【0083】図5以降で説明してきた具体例を用いて、上述の検索処理についていくつかの例を説明する。図16、図17は、2要素間の先祖・子孫関係の検索処理の一例における具体的な動作の一例の説明図である。各図においては、説明に關係する要素部分のみを示している。まず、属性SECTIONの要素の下位に本文の要素を有する文書構造を検索する場合を図16(A)に示している。この場合、属性SECTIONを有する要素は、SN<sub>o</sub>=9の要素である。また、本文はSN<sub>o</sub>=20の要素である。図10(A)及び図11から、両者とも編集単位IDが2であり、同じ部分構造(編集単位ID=2, GID=2)に属する。しかし、SN<sub>o</sub>=12の要素はこの部分構造のルートノードではない。そのため、さらにSN<sub>o</sub>=12の要素とSN<sub>o</sub>=20の要素の先祖・子孫関係をS44で判定する。すなわち、図10に示す要素情報からSN<sub>o</sub>=12のLN<sub>o</sub>が3であり、図14(C)に示す編集単位ID=2の構造情報からノードオーダーが4、最大ノードオーダーが6であることが求まる。また同様に、図11に示す要素情報(テキストテーブル)からSN<sub>o</sub>=20のLN<sub>o</sub>が6であり、図14(C)に示す編集単位ID=2の構造情報からノードオーダーが6であることが求まる。このSN<sub>o</sub>=20の要素のノードオーダー=6は、SN<sub>o</sub>=12の要素のノードオーダー=4と最大ノードオーダー=6の範囲内であるので、SN<sub>o</sub>=12の要素とSN<sub>o</sub>=20の要素とは先祖・子孫関係にあると判定される。

【0084】次に、属性DOCUMENTの要素の下位に本文の要素を有する文書構造を検索する場合を図16(B)に示している。この場合、属性DOCUMENTを有する要素は、SN<sub>o</sub>=12の要素である。また、本文はSN<sub>o</sub>=20の要素である。図10(A)及び図11から、両者とも編集単位IDが2であり、同じ部分構造(編集単位ID=2, GID=2)に属する。しかし、SN<sub>o</sub>=12の要素はこの部分構造のルートノードではない。そのため、さらにSN<sub>o</sub>=12の要素とSN<sub>o</sub>=20の要素の先祖・子孫関係をS44で判定する。すなわち、図10に示す要素情報からSN<sub>o</sub>=12のLN<sub>o</sub>が3であり、図14(C)に示す編集単位ID=2の構造情報からノードオーダーが4、最大ノードオーダーが6であることが求まる。また同様に、図11に示す要素情報(テキストテーブル)からSN<sub>o</sub>=20のLN<sub>o</sub>が6であり、図14(C)に示す編集単位ID=2の構造情報からノードオーダーが6であることが求まる。このSN<sub>o</sub>=20の要素のノードオーダー=6は、SN<sub>o</sub>=12の要素のノードオーダー=4と最大ノードオーダー=6の範囲内であるので、SN<sub>o</sub>=12の要素とSN<sub>o</sub>=20の要素とは先祖・子孫関係にあると判定される。

【0085】図16(C)に示す例では、属性CHAPTERの要素の下位に本文の要素を有する文書構造を検索する場合を示している。この場合、図10、図11に示す要素情報の検索から、属性CHAPTERを有する要素は、GID=1の部分構造(編集単位ID=1)に属するSN<sub>o</sub>=6の要素である。また、本文はGID=2の部分構造(編集単位ID=2)に属するSN<sub>o</sub>=20の要素である。両者は異なる部分構造に属する。また、図10(A)に示すノードテーブルから、SN<sub>o</sub>=6の要素は、マウントポイント数が1であり、子孫にマウントポイントが存在する。そして、図9に示すグローバル構造情報から、GID=1の部分構造とGID=2の部分構造は、先祖・子孫関係にある。これらの判定によってS47へ進み、SN<sub>o</sub>=6の要素がGID=1の部分構造のルートノードであることから、GID=1の部分構造に属するSN<sub>o</sub>=6の要素と、GID=2の部分構造に属するSN<sub>o</sub>=20の要素とは先祖・子孫関係にあると判定される。

【0086】図16(D)に示す例では、属性PARTの要素の下位に属性DOCUMENTの要素を有する文書構造を検索する場合を示している。この場合、図10、図11に示す要素情報の検索から、属性PARTを有する要素は、GID=0の部分構造(編集単位ID=0)に属するSN<sub>o</sub>=3の要素である。また、属性DOCUMENTを有する要素はGID=2の部分構造(編集単位ID=2)に属するSN<sub>o</sub>=12の要素である。また、

要素情報より、両者の編集単位 ID は異なり、両者は異なる部分構造に属する。さらに要素情報より、S N o = 3 の要素のマウントポイント数は 1 であり、子孫にマウントポイントが存在することがわかる。そして、G I D = 0 の部分構造と G I D = 2 の部分構造は、グローバル構造情報を参照することによって、先祖・子孫関係にあることがわかる。しかし、S N o = 3 の要素は、G I D = 0 の部分構造におけるルートノードではない。従って S 4 8 の判定を行うことになる。

【0087】この場合、G I D = 2 のグローバル構造情報の親 G I D から、G I D = 2 の部分構造の親は G I D = 1 の部分構造である。同様に、G I D = 1 のグローバル構造情報の親 G I D から、G I D = 1 の部分構造の親は G I D = 0 の部分構造である。このパス上で G I D = 0 の部分構造の直下の部分構造、すなわち G I D = 1 の部分構造のルートノードに対応する G I D = 0 の部分構造のマウントポイントを求める。すなわち、G I D = 1 のグローバル構造情報における親の接続 ID を取得すると、「001」となっており、属性値として E 001 を有する要素を図 11 に示す要素情報から検索する。すると、S N o = 1 6 の要素がマウントポイントであることがわかる。

【0088】さらに、図 10 (A) に示す要素情報から S N o = 3 の要素の L N o が 3 であり、図 14 (A) に示す G I D = 0 の構造情報中の L N o = 3 を参照することによって、S N o = 3 の要素のノードオーダーが 4、最大ノードオーダーが 8 であることがわかる。同様に図 11 に示す要素情報から S N o = 1 6 の要素の L N o が 8 であり、図 14 (A) に示す G I D = 0 の構造情報中の L N o = 8 を参照することにより、S N o = 1 6 のマウントポイントのノードオーダーが 8 であることがわかる。この S N o = 1 6 のマウントポイントのノードオーダー = 8 は、S N o = 3 の要素のノードオーダー = 4 と最大ノードオーダー = 8 の範囲内である。従って、S N o = 1 6 のマウントポイントと S N o = 3 の要素とは先祖・子孫関係にある。従って、G I D = 0 の部分構造に属する S N o = 3 の要素と、G I D = 2 の部分構造に属する S N o = 1 2 の要素とは先祖・子孫関係にあると判定される。

【0089】図 17 (A) に示す例では、属性 T I T L E の要素の下位に属性 D O C I T E M の要素を有する文書構造を検索する場合を示している。この場合、図 10、図 11 に示す要素情報の検索から、属性 T I T L E を有する要素は、G I D = 0 の部分構造（編集単位 ID = 0）に属する S N o = 1 の要素と S N o = 4 の要素、G I D = 1 の部分構造（編集単位 ID = 1）に属する S N o = 7 の要素、G I D = 2 の部分構造（編集単位 ID = 2）に属する S N o = 1 0 の要素の 4 つである。また、属性 D O C I T E M を有する要素は G I D = 2 の部分構造（編集単位 ID = 2）に属する S N o = 1 2 の要

素である。

【0090】このうち、S N o = 1, 4, 7 の要素については、S N o = 1 2 の要素とは異なる部分構造に属している。しかし、S N o = 1, 4, 7 の要素は、図 10 (A) に示す要素情報からマウントポイント数が 0 であり、S 4 5において S N o = 1 2 の要素との先祖・子孫関係は否定される。

【0091】また、S N o = 1 0 の要素については、S N o = 1 2 の要素と同じ部分構造（編集単位 ID = 2, G I D = 2）に属している。さらに S N o = 1 0 の要素は、この部分構造のルートノードではない。そのため、S 4 4において S N o = 1 0 の要素と S N o = 1 2 の要素の先祖・子孫関係の判定を行う。S N o = 1 0 の要素に対応するノードオーダーと最大ノードオーダーは、図 14 (C) に示す編集単位 ID = 2 の構造情報から、それぞれ 1, 3 である。また同様に S N o = 1 2 の要素に対するノードオーダーは 4 である。この S N o = 1 2 の要素に対するノードオーダー = 4 は、S N o = 1 0 の要素に対するノードオーダー = 1 と最大ノードオーダー = 3 の範囲に入っていないので、S N o = 1 0 の要素と S N o = 1 2 の要素との先祖・子孫関係は否定される。

【0092】図 5 以降で説明した具体例では、図 15 の S 4 6, S 4 8 で先祖・子孫関係が否定されることはないが、例えば図 2 に示したような文書構造の場合にあり得る。図 17 (B) には、属性「章題」の要素の下位に属性「図表」の要素を有する文書構造を検索する場合を示している。以下の説明では詳細は省略するが、属性「章題」の要素として S N o = 2 の要素が検索され、属性「図表」の要素として S N o = 1 2 の要素が検索される。この 2 つの要素は、それぞれ G I D = 0 の部分構造と G I D = 2 の部分構造に含まれており、異なる部分構造に含まれていることがわかる。また、S N o = 2 の要素の下位には、マウントポイントが存在している。さらに、G I D = 0 の部分構造と G I D = 2 の部分構造とは、先祖・子孫関係にある。従って、図 15 の S 4 8 の判定を行うが、G I D = 2 の部分構造のルートノードに対応するマウントポイントは S N o = 1 4 の要素であり、S N o = 2 とは先祖・子孫関係にはない。従って、S N o = 2 の要素と S N o = 1 2 の要素との先祖・子孫関係は否定される。

【0093】図 17 (C) に示した例では、図 2 に示した文書構造において、属性「タイトル」の要素の下位に属性「図表」の要素を有する文書構造を検索する場合を示している。やはり詳細な説明は省略するが、属性「タイトル」の要素として S N o = 3 の要素が検索され、属性「図表」の要素として S N o = 1 2 の要素が検索される。この 2 つの要素は、それぞれ G I D = 1 の部分構造と G I D = 2 の部分構造に含まれており、異なる部分構造に含まれていることがわかる。この例では S N o = 3 の要素の下位にはマウントポイントが存在していないの

で、S 4 5 の判定によって S N o = 3 の要素と S N o = 1 2 の要素との先祖・子孫関係は否定される。仮に S N o = 3 の要素の下位にマウントポイントが存在していた場合でも、G I D = 1 の部分構造と G I D = 2 の部分構造とが先祖・子孫関係にはないでの、S 4 6 の判定によって S N o = 3 の要素と S N o = 1 2 の要素との先祖・子孫関係は否定される。

【0094】以上のように、構造化文書を部分構造に分解して、部分構造間の関係を示すグローバル構造情報と、部分構造内の要素間の関係を示す構造情報を保持しているので、上述のように文書構造の検索を行うに際して、2つの要素間の先祖・子孫関係を判定する処理を、高速に実行することができる。

【0095】なお、上述の検索処理の説明では、文書のバージョンについては触れていないが、例えば検索条件としてバージョンに関する条件を付加することによって、任意のバージョンに対する文書構造の検索を行うことが可能である。また、上述の検索処理の具体例では、図4に示したデータ構造に従った検索処理の一例を示しているが、図4に示したデータ構造は一具体例であって、他のデータ構造を適用可能であり、その場合の検索処理についても本発明の趣旨を逸脱しない範囲で任意の検索手法を適用可能である。

【0096】上述の実施の形態では、構造に関する情報を部分構造間の関係を示すグローバル構造情報と要素間の関係を示す構造情報で管理構成する構成を示したが、さらに、本発明を多段階に分割したグローバル構造情報に適用することで、さらに大規模な構造化文書の検索を高速に実施できるような構成も可能である。

#### 【0097】

【発明の効果】以上の説明から明らかのように、本発明によれば、構造化文書を部分構造に分解して、部分構造間の関係を示す第1の構造情報（グローバル構造情報）と、部分構造内の要素間の関係を示す第2の構造情報

（構造情報）を保持するようにしたので、例えば文書の編集を行う場合でも、部分構造単位での編集を行うことによって、編集の影響を部分構造内にとどめることができる。そのため、例えば部分構造ごとに編集を行えば並行して複数人が編集を行うことができ、共同編集環境を提供することができる。また、編集による構造情報の再構築を部分構造内にとどめることができ、構造情報の再構築に要する時間を短縮することができる。

【0098】また、構造化文書に対して頻繁に行われる文書構造の検索に対しても、2要素間の先祖・子孫関係の判定を、部分構造間の先祖・子孫関係の判定と、先祖側の部分構造内の要素とマウントポイントとの判定を行うだけで実行できる。従って、従来のように1つずつ要

素をたぐってゆくなどの処理が簡略化され、高速に2要素間の先祖・子孫関係の判定を行って、文書構造の検索の高速化を図ることができるという効果がある。

#### 【図面の簡単な説明】

【図1】 本発明の実施の一形態を示すブロック図である。

【図2】 構造化文書の構造の一例の説明図である。

【図3】 グローバル構造情報及び要素情報と構造情報の一例の説明図である。

【図4】 具体例におけるデータ構造の一例の説明図である。

【図5】 入力される構造化文書の具体例の説明図である。

【図6】 分解された部分構造ごとの文書の具体例（編集単位 I D = 0）の説明図である。

【図7】 分解された部分構造ごとの文書の具体例（編集単位 I D = 1）の説明図である。

【図8】 分解された部分構造ごとの文書の具体例（編集単位 I D = 2）の説明図である。

【図9】 グローバル構造情報の具体例の説明図である。

【図10】 要素情報（ノードテーブル及び属性テーブル）の具体例の説明図である。

【図11】 要素情報（テキストテーブル）の具体例の説明図である。

【図12】 構造情報を生成する処理の一例を示すフローチャートである。

【図13】 編集単位 I D が 2 の部分構造における構造情報生成過程の一例の説明図である。

【図14】 構造情報及びバージョンテーブルの具体例の説明図である。

【図15】 本発明の実施の一形態における 2 要素間の先祖・子孫関係の検索処理の一例を示すフローチャートである。

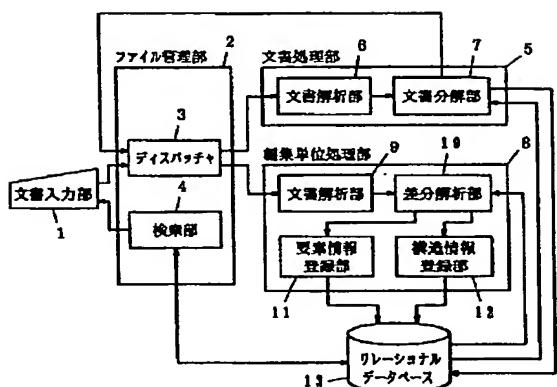
【図16】 2 要素間の先祖・子孫関係の検索処理の一例における具体的な動作（先祖・子孫関係にある場合）の一例の説明図である。

【図17】 2 要素間の先祖・子孫関係の検索処理の一例における具体的な動作（先祖・子孫関係にない場合）の一例の説明図である。

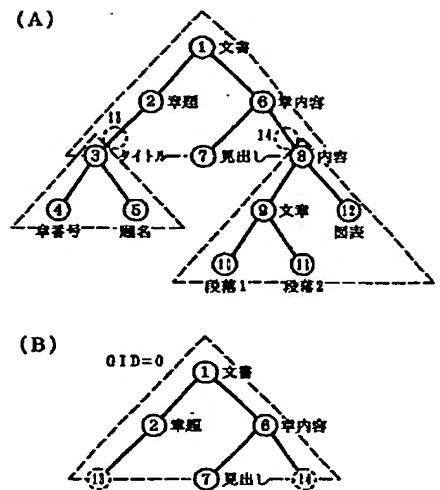
#### 【符号の説明】

1…文書入力部、2…ファイル管理部、3…ディスチャ、4…検索部、5…文書処理部、6…文書解析部、7…文書分解部、8…編集単位処理部、9…文書解析部、10…差分解析部、11…要素情報登録部、12…構造情報登録部、13…リレーションナルデータベース。

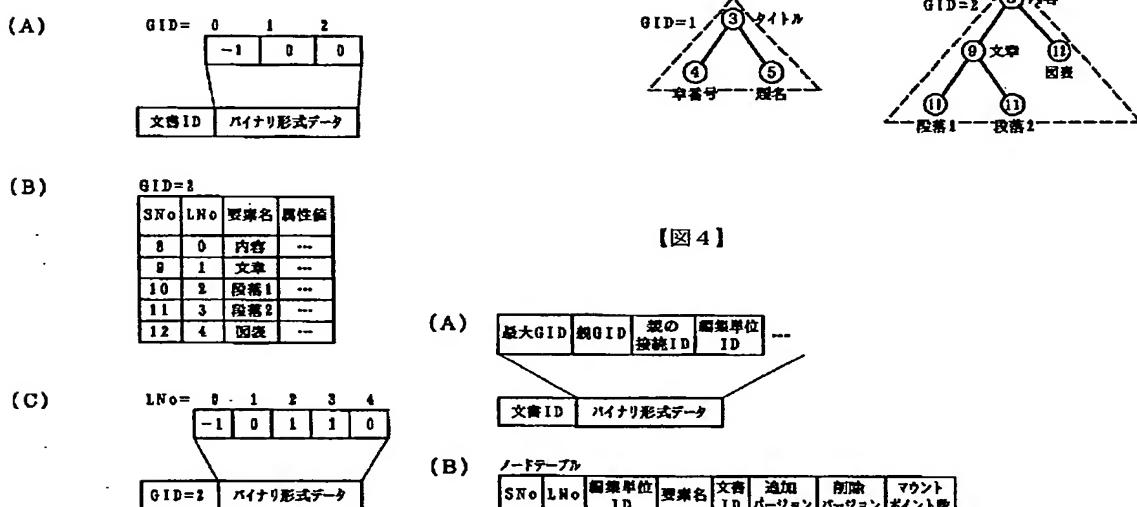
【図1】



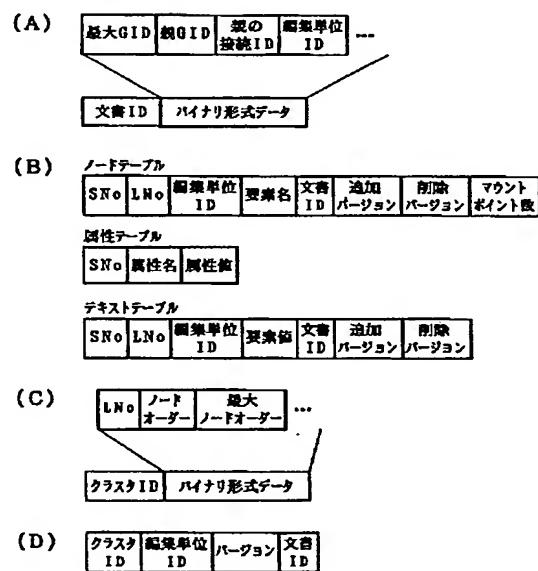
【図2】



【図3】



【図4】

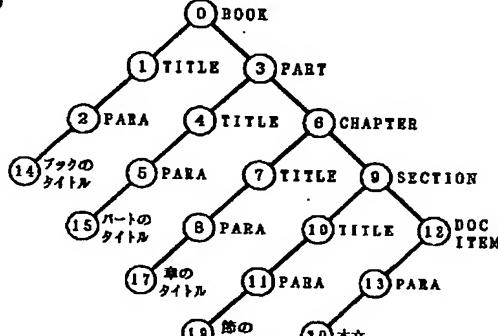


【図5】

(A)

```
(BOOK ID="7000000000000000")
(TITLE ID="7000000000000001" READING="あっくのたいとる")
(PARA ID="7000000000000002") ブックのタイトル (/PARA)
(/TITLE)
(PART ID="7000000000000003")
(TITLE ID="7000000000000004" READING="はーとのたいとる")
(PARA ID="7000000000000005") パートのタイトル (/PARA)
(/TITLE)
(CHAPTER ID="7000000000000006")
(TITLE ID="7000000000000007" READING="しょうのたいとる")
(PARA ID="7000000000000008") 章のタイトル (/PARA)
(/TITLE)
(SECTION ID="7000000000000009")
(TITLE ID="7000000000000010" READING="せつのたいとる")
(PARA ID="7000000000000011") 節のタイトル (/PARA)
(/TITLE)
(DOCITEM ID="7000000000000012")
(PARA ID="7000000000000013") 本文 (/PARA)
(/DOCITEM) (/SECTION) (/CHAPTER) (/PART) (/BOOK)
```

(B)



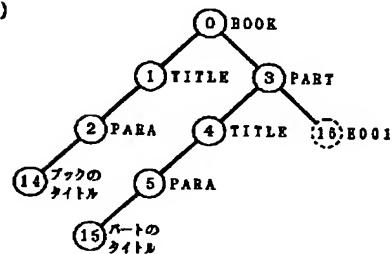
【図7】

【図6】

(A) 編集単位ID=0, GID=0

```
(BOOK ID="7000000000000000")
(TITLE ID="7000000000000001" READING="あっくのたいとる")
(PARA ID="7000000000000002") ブックのタイトル (/PARA)
(/TITLE)
(PART ID="7000000000000003")
(TITLE ID="7000000000000004" READING="はーとのたいとる")
(PARA ID="7000000000000005") パートのタイトル (/PARA)
(/TITLE) &E001; (/PART) (/BOOK)
```

(B)

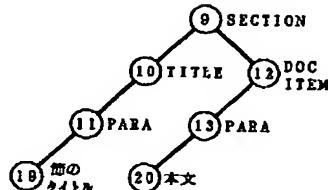


【図8】

(A) 編集単位ID=2, GID=2

```
(SECTION ID="7000000000000009")
(TITLE ID="7000000000000010" READING="せつのたいとる")
(PARA ID="7000000000000011") 節のタイトル (/PARA)
(/TITLE)
(DOCITEM ID="7000000000000012")
(PARA ID="7000000000000013") 本文 (/PARA)
(/DOCITEM) (/SECTION)
```

(B)

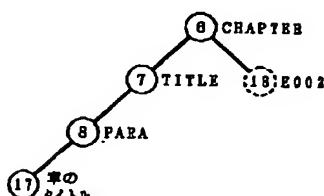


【図11】

(A) 編集単位ID=1, GID=1

```
(CHAPTER ID="7000000000000006")
(TITLE ID="7000000000000007" READING="しょうのたいとる")
(PARA ID="7000000000000008") 章のタイトル (/PARA)
(/TITLE) &E002; (/CHAPTER)
```

(B)



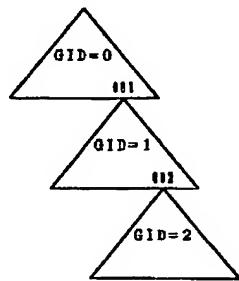
テキストテーブル

SNo	LNo	編集単位ID	要素名	文書ID	追加バージョン	既読バージョン
14	6	0	ブックのタイトル	0	1	NULL
15	7	0	パートのタイトル	0	1	NULL
16	8	E001		0	1	NULL
17	3	1	章のタイトル	0	1	NULL
18	4	1	E002	0	1	NULL
19	5	2	節のタイトル	0	1	NULL
20	6	2	本文	0	1	NULL

【図9】

(A) GID=0	GID=1	GID=2
2 : -1 : -1	0 : 2 : 0	01 : 1 : 2 : 1 : 001 : 2

(B)



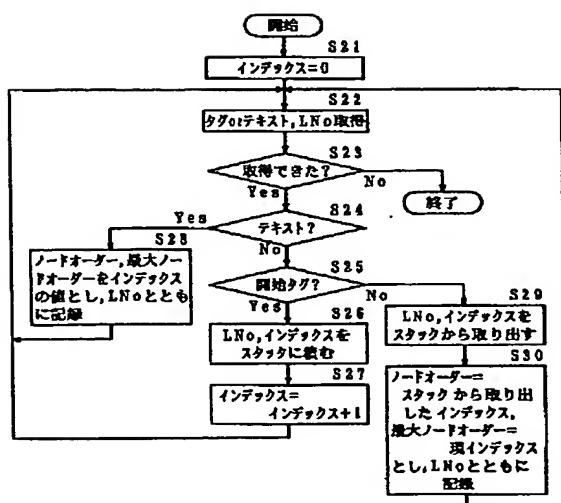
【図10】

(A) ノード テーブル		記録単位 ID	要素名	文書 ID	追加 バージョン	削除 バージョン	マウント ポイント数
0	0	BOOK	0	1	NULL	1	
1	1	TITLE	0	1	NULL	0	
2	2	PARA	0	1	NULL	0	
3	3	PART	0	1	NULL	1	
4	4	TITLE	0	1	NULL	0	
5	5	PARA	0	1	NULL	0	
6	0	CHAPTER	0	1	NULL	1	
7	1	TITLE	0	1	NULL	0	
8	2	PARA	0	1	NULL	0	
9	0	SECTION	0	1	NULL	0	
10	1	TITLE	0	1	NULL	0	
11	2	PARA	0	1	NULL	0	
12	3	DOCITEM	0	1	NULL	0	
13	4	PARA	0	1	NULL	0	

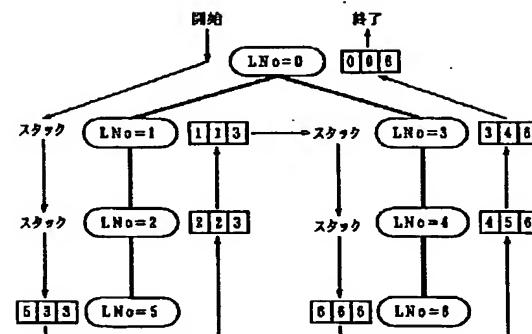
(B)  
属性  
テーブル

SNo	属性名	属性値
0	ID	7000000000000000
1	ID	7000000000000001
2	READING	あっくのたいとる
3	ID	7000000000000002
4	ID	7000000000000003
4	READING	ばーとのたいとる
5	ID	7000000000000004
6	ID	7000000000000005
7	ID	7000000000000006
7	READING	しょうのたいとる
8	ID	7000000000000007
9	ID	7000000000000008
10	ID	7000000000000009
10	READING	せつのたいとる
11	ID	7000000000000010
12	ID	7000000000000011
13	ID	7000000000000012

【図12】



【図13】



【図 1 4】

(A) 編集単位ID=0, クラスタID=0

LNo	ノードオーダー	最大ノードオーダー
0	0	8
1	1	8
2	2	3
3	4	8
4	5	7
5	6	1
6	3	8
7	7	7
8	0	8

(B) 編集単位ID=1, クラスタID=1

LNo	ノードオーダー	最大ノードオーダー
0	0	4
1	1	3
2	2	3
3	3	3
4	4	4

(C) 編集単位ID=2, クラスタID=2

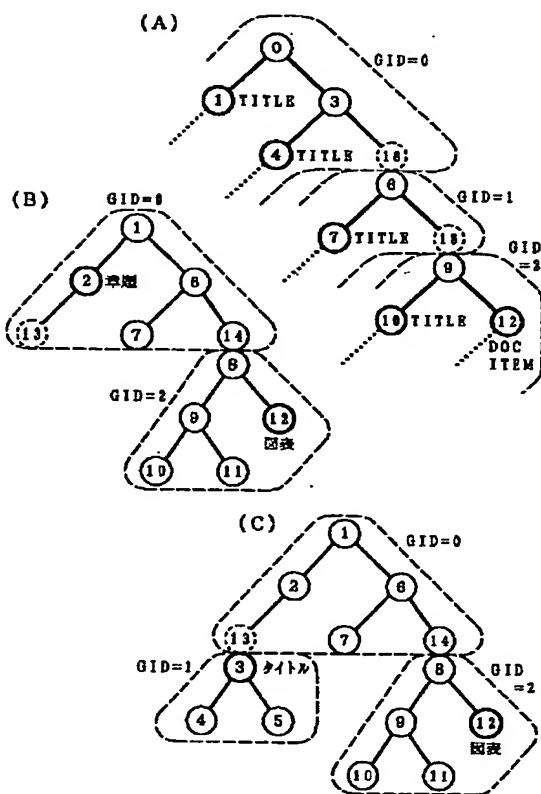
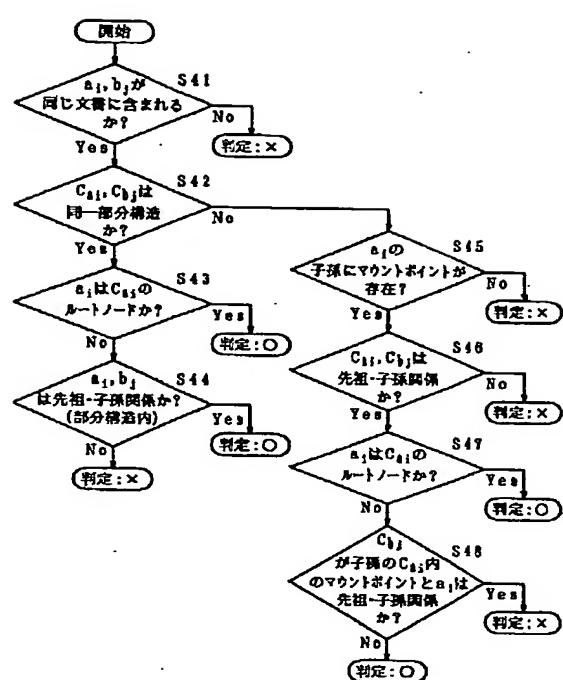
LNo	ノードオーダー	最大ノードオーダー
0	0	6
1	1	3
2	2	3
3	4	6
4	5	6
5	3	3
6	6	6

(D) バージョンテーブル

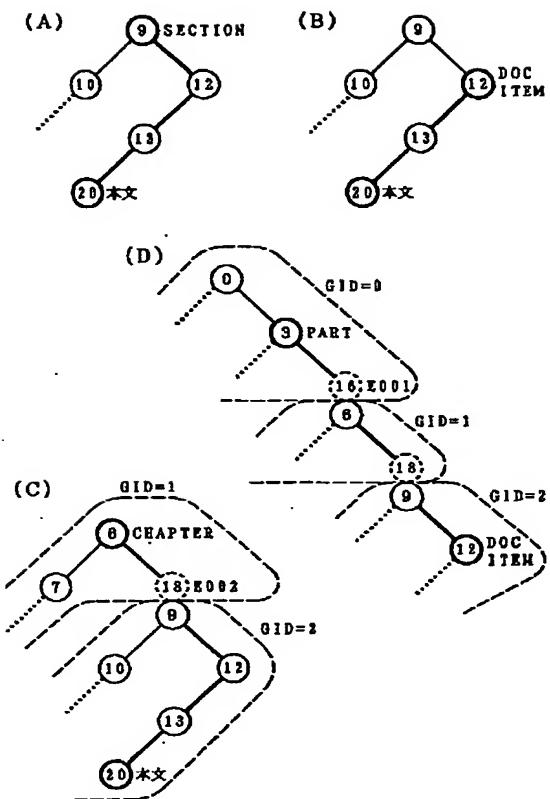
クラスタID	編集単位ID	バージョンID	文書ID
0	0	1	0
1	1	1	0
2	2	1	0

【図 1 7】

【図 1 5】



【図16】



フロントページの続き

(72) 発明者 領賀 雅夫 神奈川県横浜市西区みなとみらい3丁目3 番1 富士ゼロックス株式会社内	(72) 発明者 東 和彦 神奈川県横浜市西区みなとみらい3丁目3 番1 富士ゼロックス株式会社内
(72) 発明者 山田 季史 神奈川県横浜市西区みなとみらい3丁目3 番1 富士ゼロックス株式会社内	(72) 発明者 山田 美穂 神奈川県横浜市西区みなとみらい3丁目3 番1 富士ゼロックス株式会社内
(72) 発明者 池田 稔 神奈川県横浜市西区みなとみらい3丁目3 番1 富士ゼロックス株式会社内	Fターム(参考) 5B009 NA05 SA03 SA12 5B075 ND36 NK44 NK54 QT06